Step 14 – JDBC – Some extra care

Time to work with databases in something more of a realistic setting. Time to ensure that all our changes can be rolled back, and that we are protected against SQL injection.

The goals for this step:

- Handle transaction commitment and rollback.
- Protect your project against SQL injections, by using PreparedStatements.
-

Task 1 – Shapes (but safe).

Expanding previous work. You will start from your own Shapes with MoveablePoints solution, or using the one in the exercises/ex13/shapes. This includes your previous work supporting database connections.

The task is to create a class that allows you to create new shapes and add them to a database. You should:

- Have a method that adds a shape to a database. The shape and moveable points will be added at the same time. Only valid objects will be added (so, with the appropriate points, all relevant fields set, etc. Otherwise, the transaction will be rolled back.
- Have a method that finds an object in a database, based on some parameter. Make sure that the method is not vulnerable to SQL injection.
- Have a method that returns all objects from a database
- Have a method that updates the position of several shapes at the same time (e.g. a group of objects is moved in a given direction for a given distance at the same time, and the new positions are updated in the DB).

Are there any operations that might be interesting to add??

Task 2 – (Safe) Database support

This task is based on your solution to the first large task (or you can use the version in ex13/books). This will expand on the database support offered in the previous exercise.

Instead of storing the books in a text file, offer the same functionality, but using databases instead. Provide commit and rollback support. Make sure your methods are safe from SQL injection.

Task 2b – Expansion

Once you have the previous point running smoothly, it is time to increase complexity a bit. Instead of having the authors as Strings, create a class Author that represents that.

Again, provide safe support for database connections.

An author may have new books, planned books, etc. Store those as separate Collections of Books (ArrayList, HashMap, whatever you prefer).

Make sure your implementation has a means of allowing the authors to edit the lists of finished and ongoing books.

Make sure that a book can only be finished OR ongoing, so when updating the titles associated with an author, all modifications must be simultaneous.